IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Application of:	Group Art Unit: 2167
Bobrovskiy, Stanislav M., et al.	Confirmation No. 3079
Application No.: 10/736,452	Examiner: Lu, Kuen S.
Filed: December 14, 2003	
For: METHOD AND APPARATUS FOR BUFFERING STREAMING MEDIA	

APPELLANTS' BRIEF

TO THE COMMISSIONER FOR PATENTS:

This communication is submitted in response to the Final Office Action dated March 19, 2007 ("Office Action") and the Notice of Appeal filed on June 21, 2007. This brief pertains to the captioned patent application identified above. This Brief is being filed under the provisions of 37 C.F.R. § 41.37. The Filing Fee corresponding to this Appeal Brief, as set forth in 37 C.F.R. § 41.20(b)(2), accompanies this communication.

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Federal Cases

Federal Statutes

I. REAL PARTY IN INTEREST

The rights of the inventors in this application have been assigned to RealNetworks, Inc. of Seattle, Washington by way of assignment from Stanislav Bobravskiy and Jeffrey Chasen ("Bobravskiy et al") who are the named inventors and are captioned in the present brief.

II. RELATED APPEALS AND INTERFERENCES

Appellants, Appellants' legal representative, and the above-identified assignee are unaware of other appeals or interferences which will directly affect or be directly affected by or have a bearing on the Board's decision in the present appeal.

III. STATUS OF THE CLAIMS

Claims 1-24 are pending. Appellants appeal the rejection of each of these claims.

IV. STATUS OF AMENDMENTS

A full set of claims as currently entered is attached in Appendix A. No amendments were filed subsequent to final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

A. Claim 1 (and related Claim 13)

Independent Claim 1 recites:

> A method of storing streamed presentation data within a container file, the method comprising:
>
> receiving one or more data streams from each of one or more presentation sources within the presentation;
>
> creating within the container file, a virtual file for each of the one or more presentation sources;
>
> temporarily storing first data associated with a first data stream of a first presentation source in association with a first virtual file corresponding to the presentation source;
>
> determining a container file size of the container file;
>
> temporarily storing additional data from the first data stream in place of at least a portion of the first data if the container file size is within a predetermined range of an identified maximum buffer size; and
>
> rendering at least one of said one or more data streams.

Claim 13 claims a machine readable medium having machine executable instructions, which when executed operate to implement a method comprising the same steps that are recited in Claim 1.

Claims 1 and 13 find general support in the specification in the description of Fig. 1

in Paras. [18–20]. Specifically, these recitations find support as follows. Para. [18], and Fig. 1 disclose that as "shown digital data 102, containing data associated with one or more of presentation 103(a-c), is received by client," which supports the recitation, "receiving one or more data streams from each of one or more presentation sources within the presentation." Para. [19] discloses that a "virtual file may store data associated with one or more streams of a particular presentation source," which supports the recitation, "creating within the container file, a virtual file for each of the one or more presentation sources." Para. [19], states that "record source 1 (110) may store data associated with stream 1 of presentation source 1 (103a) within a first virtual file," which supports the recitation, "temporarily storing first data associated with a first data stream of a first presentation source in association with a first virtual file corresponding to the presentation source." Next, Para. [20] discloses that "buffer size may be specified in terms of an amount of content" or "an amount of time" or that "a user may specify the maximum buffer size," which supports the recitation, "determining a container file size of the container file." Para. [20] discloses that "if the container file reaches a specified maximum buffer size, core logic 104 may loop within the container file such that additional received data of a first presentation source may be stored in place of previously stored data," which supports the recitation of "temporarily storing additional data from the first data stream in place of at least a portion of the first data if the container file size is within a predetermined range of an identified maximum buffer size." Finally, Para. [19] discloses "passing the data to one or more decoders/renderers," which supports the recitation, "rendering at least one of said one or more data streams."

Thus, pending Claims 1 and 13 recite a method of storing streamed presentation data within a container file in a unique manner that is not taught or suggested by the prior art of record. Specifically, the claimed method that includes provisions for a virtual file for each of one or more presentation sources, along with a determination of container file size.

B. Claims 2-12 and 14-24

Claims 2-12 and 14-24 depend from independent claims and include all of the pertinent recitations discussed above.

Claims 2 and 14 further recite:

> wherein the additional data from the first data stream is stored in place of at least a portion of the first data if the container file size is equal to or exceeds the identified maximum buffer size.

These recitations are depicted in Fig. 1 and described in Para. [23], which states that "additional data may be stored in association with the corresponding record source" where the "container file size is not within a predetermined range of an indicated maximum buffer size." Pending claims 2 and 14 describe the re-use of space in a container file, which facilitates continued buffering when container file capacity has been reached or exceeded.

Claims 3 and 15 further comprise:

> temporarily storing second data associated with a second data stream of the first presentation source in association with the first virtual file; and temporarily storing additional data from the second data stream in place of at least a portion of the second data stored in association with the first virtual file if the container file size is within the predetermined range of the identified maximum buffer size.

These recitations are depicted in Fig. 1 and described in Para. [19], which states that "record source 1 (110) may store data associated with stream 1 of presentation source 1 (103a) within a first virtual file, whereas record source (111) may store data associated with stream 1 and stream 2 of presentation source 2 (103b) within a second virtual file." Pending claims 3 and 15 describe the re-use of space in a container file, which facilitates continued buffering of two data streams associated with presentation source when container file capacity has been reached or exceeded.

Claims 4 and 16 depend from Claims 3 and 15, respectively, and further comprise:

> rendering one of the first and second data streams in real-time contemporaneous with the storing of at least one of the first and second data streams.

These recitations are depicted in Fig. 5 and described in Para. [36], which states that "contemporaneous rendering/playback of each of the record sources may begin while newly received data continues to be buffered." Pending claims 4 and 16 describe the rendering of two data streams while buffering of one or both of the data streams.

Claims 5 and 17 depend from Claims 3 and 15, respectively, and further comprise:

> temporarily storing data associated with a third data stream of a second presentation source in association with a second virtual file; and temporarily storing additional data from the third data stream in place of at least a portion of the data stored in association with the second virtual file if the container file size is within the predetermined range of the identified maximum buffer size.

These recitations are illustrated in Fig. 1 and its associated text. For example, Para. [19] states, "each source-specific virtual file may store data associated with one or more streams of a particular presentation source" and "each source-specific virtual file may store data associated with one or more streams of a particular presentation source," which provides support for "storing data associated with a third data stream of a second presentation source in association with a second virtual file," as claimed in Claims 5 and 17. Para. [20] goes on to state that "if the amount of data stored within the container file falls within a predetermined range of a specified maximum buffer size, additionally received data may be stored in place of data previously stored in the container," which provides support for "storing additional data from the third data stream in place of at least a portion of the data stored in association with the second virtual file if the container file size is within the predetermined range of the identified maximum buffer size," as claimed in Claims 5 and 17.

Claims 6 and 18 further recite:

> wherein the maximum buffer size is proportional to an amount of time indicated via a user interface.

These recitations are described in Para. [20], which states,

> a maximum buffer size that may be stored within the container file may be specified. The buffer size may be specified in terms of an amount of content (e.g., 24MB) or an amount of time (e.g., 30 Minutes). In one embodiment, a user may specify the maximum buffer size that may be stored within the container file via a graphical user interface.

Claims 7 and 19 further recite:

> wherein the maximum buffer size is dynamically increased during the storing of data from the first data stream.

These recitations are described in Para. [22], which specifies that during storing of presentation data ("data from the first data stream"), the container file (which buffers the

incoming data) is initialized ("dynamically increased"), thereby determining the "maximum buffer size."

Claims 8 and 20 further recite:

> wherein the first data and additional data are stored in a native packet format prior to a decoding process.

These recitations are illustrated in Fig. 2 and its associated text. Specifically, Para. [21] states, "data is received in the form of time-interleaved packets corresponding to one or more presentation sources. The presentation data may represent a number of data types...." Block 208 indicates that the received data is stored in a record source/virtual file, and Fig. 1 illustrates that data is stored in such record sources prior to being rendered (decoded).

Claims 9 and 21 further recite:

> wherein each virtual file comprises: at least a first data block; and a file descriptor block containing at least a seek index and a seek index granularity, wherein the seek index indicates a plurality of equally distributed data blocks within the corresponding virtual file and the granularity indicates a size for each of the data blocks.

These recitations are described in Fig. 3, which illustrates a virtual file comprising a first data block 320a and 322a, a file descriptor block containing at least a seek index and a seek index granularity 316a and 318a. Para. [27] states that "the seek index represents equally distributed data blocks within a corresponding virtual file, and the seek index granularity indicates the size of each of the indexed data blocks in the virtual file," providing support for "wherein the seek index indicates a plurality of equally distributed data blocks within the corresponding virtual file and the granularity indicates a size for each of the data blocks," as claimed in Claims 9 and 21.

Claims 10 and 22 depend from Claims 9 and 21, respectively, and further recite:

> wherein the additional data is stored in place of the first data beginning with the first data block and continuing with successive data blocks of the first virtual file.

These recitations are described in Para. [30], which states that "the next data block to be written in the virtual file may be stored in place of or written over at least a portion of existing data associated with the same presentation source... such that the data effectively

loops within the current virtual file."

Claims 11 and 23 depend from Claims 9 and 21, respectively, and further recite:

> wherein if the container file size is within the predetermined range of the identified maximum buffer size, the seek index granularity is increased so as to increase data block size without changing the number of seek index entries.

These recitations are described in Para. [27], which states as follows:

> the seek index represents equally distributed data blocks within a corresponding virtual file, and the seek index granularity indicates the size of each of the indexed data blocks in the virtual file. In one embodiment, if the size of the container file falls within a determined range of an indicated maximum buffer size, the seek index granularity is increased.

Claims 12 and 24 depend from Claims 9 and 21, respectively, and further recite:

> receiving a user indication identifying a location corresponding to a time (T) within the presentation; identifying a seek position for each virtual file, wherein each seek position is determined by dividing time (T) by the seek granularity for the corresponding virtual file; and contemporaneously rendering in real-time, data stored in each virtual file at the respective seek positions.

These recitations are supported by Fig. 5, which illustrates "receiving a user indication identifying a location corresponding to a time (T) within the presentation" 502; "identifying a seek position for each virtual file, wherein each seek position is determined by dividing time (T) by the seek granularity for the corresponding virtual file" 510; and "contemporaneously rendering in real-time, data stored in each virtual file at the respective seek positions" 516.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The issues in this appeal are as follows:

1. Whether Claims 1-24 are obvious in light of Published U.S. Patent application No. 2003/0110504 to Plourde et al. in view of Korst's U.S. Patent No. 6,205,525, wherein neither *Plourde* nor *Korst* teaches or suggests, alone or in combination, a "container file" as claimed in independent Claims 1 and 13.

2. Whether one of ordinary skill in the art would have been motivated to combine the teachings of *Plourde* with those of *Korst*, wherein the problems to be solved as set out in the background sections of *Korst* and *Plourde* relate at most only peripherally to the problems addressed in Claims 1-24, and wherein a combination of *Korst* and *Plourde* would not yield the results obtained in Claims 1-24.

VII. ARGUMENT

Issue 1: Claims 1-11 and 13-23 are not obvious in light of *Plourde* in view of *Korst*.

A1. Claims 1-11 and 13-23 have been rejected under 35 U.S.C. § 103(a) as being obvious in light of U.S. Patent application No. 2003/0110504 to Plourde et al. (hereinafter "*Plourde*") in view of Korst's U.S. Patent No. 6,205,525 (hereinafter "*Korst*"). However, Appellants respectfully submit that the § 103 rejections in the Office Action are based on clearly erroneous errors, including the error that a "container file" as recited in Claims 1 and 13 is the equivalent of a hard drive with a File Allocation Table ("FAT"), as disclosed by *Plourde*.

Claims 1 and 13

Plourde discloses a Digital Video Recorder system that allows storing of subscriber television content. *Korst* discloses a video on demand system that retrieves stored content and streams it to a user device. Neither *Plourde* nor *Korst*, alone, or in combination, teach or suggest a method of storing streamed presentation data within a container file that includes a virtual file for each of one or more presentation sources. In contrast, amended Claim 1 reads as follows:

> A method of storing streamed presentation data within a container file, the method comprising:
>
> receiving one or more data streams from each of one or more presentation sources within the presentation;
>
> creating within the container file, a virtual file for each of the one or more presentation sources;
>
> temporarily storing first data associated with a first data stream of a first presentation source in association with a first virtual file corresponding to the

presentation source;
determining a container file size;
temporarily storing additional data from the first data stream in place of at least a portion of the first data if the container file size is within a predetermined range of an identified maximum buffer size; and
rendering at least one of said one or more data streams.

Claim 13 claims a machine readable medium having machine executable instructions, which when executed operate to implement a method comprising the same steps that are recited in Claim 1.

It is apparent from Claims 1 and 13 that a "container file" has at least two necessary attributes, neither of which is taught or suggested by the prior art. First, a "container file" must vary in size, otherwise there would be no need to "determine a container file size of the container file." Second, the contents of a container file, the "virtual file for each of the one or more presentation sources," are dynamic and transient, storing **temporarily** data from a data stream until it can be rendered.

Plourde, by contrast, discloses a Digital Video Recorder ("DVR") system that allows the long term storage of subscriber television content on a storage device, preferably a hard drive (*Plourde* Para. 87). *Plourde* teaches that the storage device is of a fixed size and that the operating system makes use of a File Allocation Table ("FAT") to store information about the hard disk clusters and the files associated with those clusters (*Plourde* Para. 88).

On page 3, the Office Action suggests that a "container file," as in Claims 1 and 13, is taught by *Plourde's* hard drive that contains information about media content instance files in a FAT file. However, a container file, as in Claims 1 and 13, offers a level of flexibility that is not needed, nor taught or suggested by *Plourde's* hard drive with a FAT file. As demonstrated by the attributes discussed above, the size of a container file may be adapted to suit various circumstances, while a hard drive is not flexible enough to do so, being by its nature limited to a fixed size. In addition, the creation of *Plourde's* hard drive with a FAT file typically requires destructively "formatting" the hard drive. By contrast, container files, as in Claims 1 and 13, can be easily and nondestructively created and disposed of as needed. Furthermore, because a FAT is a type of file system, *Plourde's* hard drive would typically involve a file system's overhead and would be complex to design and administer. By contrast, "container files" offer facilities to perform few complex tasks and are therefore not

only more flexible than *Plourde's* hard drive, but also much simpler. Accordingly, it is clear error to say that *Plourde* teaches this element of Claims 1 and 13.

The Office Action further suggests that a "virtual file for each of one or more presentation sources," as in Claim 1, is taught by *Plourde's* media content instance files. In support of this suggestion, the Office Action on page 11 states that "the FAT [has] entries describing attributes of content media instance files where directory structured virtual file contains one or more entries and the structure does teach one directory having one entry." Apparently, the argument is that a "virtual file for each of one or more presentation sources" is identical to a disk drive with a file allocation table that contains an entry describing a single file. Appellants respectfully submit that the "virtual file for each of one or more presentation sources" is much more flexible and useful than the scenario described in the Office Action. Appellants further respectfully submit that *Plourde* lacks any suggestion that media content instance files may be virtually aggregated and organized by presentation source. Such convenience is offered only by Claims 1 and 13; therefore, it is clear error to say that *Plourde* teaches this element of these claims.

The clear errors, however, are not limited to those aspects of Claims 1 and 13 that are said to be taught by *Plourde.* If anything, the errors regarding *Korst* are even clearer. For example, after correctly noting that *Plourde* does not teach "determining a container file size," the Office Action suggests on pages 3-4 that "determining a container file size," as in Claims 1 and 13, is taught by *Korst's* calculation of the size of the data block to be read in the following sweep operation. However, while *Korst* may disclose a process that involves a "determination" of a "size," even a cursory reading of *Korst* makes clear that it does not teach "determining a *container file* size." A brief summary of *Korst* explains why.

Korst discloses a video on demand *server* that efficiently buffers content data for streaming to a user device. The content data addressed in *Korst* is stored on a storage medium, such as a hard drive, and must be retrieved from the storage medium in blocks (*Korst* col. 7, lines 13-42). The calculation that *Korst* teaches involves figuring out how many blocks of data to read from the hard drive and insert into a buffer for ultimate delivery to a client device, a calculation that depends not on the contents of any container file, but on the number of streams that the server is sending out to client devices (*Korst* col. 7, lines 13-

42). Thus, it is clear error to say that *Korst* teaches "determining a container file size" as in Claims 1 and 13 because unlike Claims 1 and 13, *Korst* **does not determine the size of any sort of container**.

Finally, it is clear error to say that "rendering at least one of said one or more data streams" is taught by *Korst's* scheduler's determining the number of active streams and calculating the size of the data block to be read, as suggested by the Office Action on page 4. The cited reference to *Korst* does not teach or even suggest the "rendering" of a stream; rather, *Korst* teaches at most only that a data stream may be transmitted to a client, a process that is very different from "rendering" a data stream into a form that may be perceived by a user.

For the reasons just discussed, Appellants respectfully request that the Board overturn the rejections of Claims 1 and 13.

Claims 2-11 and 14-23

A2. Claims 1 and 13 are allowable as noted above. Additionally, dependent Claims 2-11 and 14-23 are allowable because they depend from allowable claims. These claims include further recitations not taught, disclosed, or even suggested by *Plourde* and/or *Korst*. A nonexclusive listing of some additional reasons Claims 2-11 and 14-23 are allowable are included below.

For example, Claims 3 and 15 recite "storing second data associated with a second data stream of the first presentation source in association with the first virtual file." *Plourde* does not even mention any reference to virtual files. *Plourde* merely cites a conventional FAT storage system with convention separate files on a storage medium. In particular, *Plourde* (and *Korst*) fail to teach or suggest storing a presentation that can contain multiple sources within a single container file as one or more virtual files. For these reasons, in addition to those already noted above, Claims 3 and 15 and their dependent claims are in condition for allowance, and the Office Action erroneously concluded to the contrary. Appellants therefore respectfully request that the Board overturn these rejections.

A3. In another example, Claims 9 and 21 include "a first data block; and a file descriptor block containing at least a seek index and a seek index granularity, wherein the

seek index indicates a plurality of equally distributed data blocks within the corresponding virtual file and the granularity indicates a size for each of the data blocks." *Plourde* and *Korst* also fail to teach and/or suggest such a seek index, let alone a seek index that indicates a plurality of equally distributed data blocks within the corresponding virtual file. The portions of *Plourde* that were cited as teaching a seek block (paragraphs 88 and 110) do not mention seek indexes, let alone a seek index as recited in Claims 9 and 21. *Plourde* merely states "The type of media content (e.g. westerns, comedies, action, etc) can be presented to the user (for selection, or user configurable without a pre-configured list), and then a preference filter can seek and effect the receipt of such content for contemporaneous and/or later viewing." It is clear that *Plourde's* mere mention of the word "seek" is not sufficient a teaching to render Claims 9 and 21 obvious.

For this reason as well, in addition to those already noted above, Claims 9 and 21 and their dependent claims are in condition for allowance. The Office Action has not established that Claims 9 and 21 are obvious in light of *Plourde* and/or *Korst,* and the Appellants, therefore respectfully request that the Board overturn these rejections.

Issue 2: One of ordinary skill in the art would not have been motivated to combine *Plourde* and *Korst*.

B1. The Office Action erred in determining that one of ordinary skill in the art would have been motivated to combine *Plourde* and *Korst* for at least two reasons. First, the problems that *Korst* and *Plourde* solve relate at most only peripherally to the problems addressed in Claims 1-24. Second, the Examiner used unacceptable hindsight reasoning to determine that Claims 1-24 are supposedly obvious in light of *Korst* and *Plourde*.

It is well established that the prior art must suggest the desirability of the claimed invention. MPEP 2143.01. Obviousness can only be established where there is some teaching, suggestion, or motivation to combine or modify the teachings of the prior art to produce the claimed invention. *In re Kahn,* 441 F.3d 977, 986, 78 USPQ2d 1329, 1335 (Fed. Cir. 2006). *See also KSR Intern. Co. v. Teleflex Inc.,* 127 S.Ct. 1727, 1741 (2007) (explaining that it is generally "important to identify a reason that would have prompted a person of ordinary skill in the relevant field to combine the elements in the way the claimed new invention does"). It is equally well established that the motivation-suggestion-teaching

requirement exists to guard against using hindsight in an obviousness analysis. *Id.*

The Office Action stated an ordinarily skilled practitioner's motivation to combine *Plourde* and *Korst* as follows:

> It would have been obvious to one having ordinary skill in the art at the time of the, applicant's invention was made to combine the teaching of Korst with Plourde reference by tracking duration of time shift buffer and calculating buffering rate to determine the buffer capacity size because both references are directed to media content delivery and the combined teaching of the references would have made Plourde's buffering mechanisms for video recording and delivery more efficient due to more accurately calculated and determined buffer size.

On page 10 of Appellants' response of October 17, 2006, the Appellants noted that "the Examiner has attempted to use the pending application to define the problem to be solved by reference to different elements from the prior art." On page 12 of the Office Action, the Examiner replied that "the motivation or suggestion of combination of the references does come from the BACKGROUNDS OF INVENTION of the references...."

The background sections of *Korst* and *Plourde* do not, counter to the Examiner's argument, provide a motivation to combine. Generally, the background section of a patent merely lays out the problem to be solved. Because the problems to be solved that are laid out in the background sections of *Korst* and *Plourde* relate at most only peripherally to the problems that the Appellants address, and because there is no suggestion in either background to make such a combination, Appellants respectfully submit that the background sections of *Korst* and *Plourde* provide no motivation to combine.

For example, *Plourde* explains that it addresses the problems that arise when a television viewer wishes to watch two or more programs at the same time or wishes to watch a program that is on when the viewer is away from the television (*Plourde* para 5). *Plourde* solves these problems by providing a better "buffering" mechanism that would be employed by a digital video recorder, such as a Tivo®. In other words, *Plourde* works with media data that arrives from elsewhere at a constant rate, at a known time, for a known duration.

By contrast, *Korst* explains that it addresses problems faced by streaming media servers that may have to stream many different streams of different pieces of media out to many different clients. To that end, *Korst* provides, "[t]o supply data to a user as a continuous data stream, special scheduling schemes for reading data from the disks are required with an appropriate scheme for temporarily buffering the read data before the data is

supplied to the user" (*Korst* col. 1, lines 55-58). In other words, *Korst* works with a large quantity of media data that is stored locally, but that distant, disparate clients may want to access at unknown rates, at unknown times, for unknown durations.

Thus, as their background sections make clear, the **only** commonalities between *Korst* and *Plourde* are that they both relate to media data sent from one place to another. On the other hand, there are many differences that would have deterred one of ordinary skill in the art from combining the two. Particularly, *Korst* addresses problems that are not faced by media-receiving client devices, such as are the subject of *Plourde*. Unlike streaming media servers, which are the subject of the problems solved by *Korst,* a client device has no need to read data in sweeps from a hard drive; a client device does not need to service multiple requests for data streams; nor does a client device need to juggle the delivery of different pieces of media that are stored in disjunct sectors on a local hard drive. Clearly, the problems faced by streaming servers differ greatly from those faced by client devices. Therefore, there is no reason why a person of ordinary skill in the art would have been motivated to combine *Plourde*, which deals with client devices, and *Korst*, which deals with server devices.

B2. Appellants further respectfully submit that any teaching, suggestion, or motivation to combine *Plourde* with *Korst* can be found only by using the Application as a blueprint for piecing together individual claim components in hindsight. By analyzing the inventions claimed in Claims 1-24, the Office Action has identified a particular set of components, including (A) "temporarily storing additional data from the first data stream in place of at least a portion of the first data if the container file size is within a predetermined range of an identified maximum buffer size" and (B) "rendering at least one of said one or more data structure." *Plourde* is said to disclose A, but not B; whereas *Korst* is said to disclose B, but not A. However, there is no suggestion in *Plourde* that B would be a desirable addition to the invention described in *Plourde*. Nor is there any suggestion in *Korst* that A would be a desirable addition to the invention described in *Korst*.

Therefore, at the time the inventions claimed in Claims 1-24 were made, there was nothing to suggest to an ordinarily skilled practitioner that there was any need to combine A with B in order to produce (in part) the inventions claimed in Claims 1-24. To the contrary, only since Appellants invented the inventions claimed in Claims 1-24 is it possible to look back and attempt to piece together components from *Plourde* and *Korst* using Claims 1-24 as

a template. But such a retrospective component parts analysis is expressly forbidden:

> In making the assessment of differences, section 103 specifically requires consideration of the claimed invention "as a whole." Inventions typically are new combinations of existing principles or features. The "as a whole" instruction in title 35 prevents evaluation of the invention part by part. Without this important requirement, an obviousness assessment might break an invention into its component parts (A + B + C), then find a prior art reference containing A, another containing B, and another containing C, and on that basis alone declare the invention obvious. This form of hindsight reasoning, using the invention as a roadmap to find its prior art components, would discount the value of combining various existing features or principles in a new way to achieve a new result-often the very definition of invention.

Ruiz v. A.B. Chance Co., 357 F.3d 1270, 1275 (Fed. Cir. 2004) (internal citations omitted); *see also, e.g., Texas Instruments Inc. v. U.S. Intern. Trade Com'n*, 988 F.2d 1165, 1178 (Fed. Cir. 1993) (holding that it is impermissible to "piece the invention together using the patented invention as a template" when the "references in combination do not suggest the invention as a whole"); *Envtl. Designs, Ltd. v. Union Oil Co.*, 713 F.2d 693, 698 (Fed. Cir. 1983) (noting that "**virtually all [inventions] are combinations of old elements**").

Given the differences between the subject matters disclosed by *Korst* and *Plourde*, given the differences between the problems solved by *Korst* and *Plourde*, and given the lack of any suggestion in either *Korst* or *Plourde* that there would be a benefit to adding the teaching of one to the other, the only way that one could combine *Korst* and *Plourde* to reach the results obtained in the recited Claims is by "engage[ing] in a hindsight reconstruction of the claimed invention, using the applicant's structure as a template and selecting elements from references to fill the gaps." *See In re Gorman,* 993 F.2d 982, 18 U.S.P.Q.2d 1885 (1991). And that is exactly what the Examiner has done. However, the Examiner may not "use hindsight reconstruction to pick and choose among isolated disclosures in the prior art" to determine that the recited Claims are unpatentable over *Plourde* in view of *Korst. See Ecolochem, Inc. v. Southern California Edison Co.*, 227 F.3d 1361, 56 U.S.P.Q.2d 1065 (2000).

Accordingly, Appellants respectfully submit that the Office Action did not state a *prima facie* case of obviousness for Claims 1-24. Appellants further request reconsideration and withdrawal of the rejections of Claims 1-24.

VIII. SUMMARY

In view of the foregoing, Appellants respectfully request reconsideration and withdrawal of the rejection of amended Claims 1 and 13. In addition, Appellants submit that Claims 2-11 and 14-23, which depend directly or indirectly on Claims 1, and 13, are patentably distinct over the combination of *Plourde* in view of *Korst*.

Appellants therefore submit that all pending claims are in condition for allowance. Accordingly, early and favorable action allowing all of the pending claims and passing this application to issue is respectfully requested. The Board is invited to contact the undersigned at the telephone number below if there are any remaining questions regarding this application.

We believe the appropriate fees accompany this transmission. If, however, insufficient fee payment or fee overpayment occurs, the amount may be withdrawn or deposited from/to Axios Law Group's deposit account. The deposit account number is 50-4051.

Respectfully submitted,
AXIOS LAW GROUP

Date: September 21, 2007

by: /Adam L.K. Philipp/
Adam L.K. Philipp
Reg. No.: 42,071
Direct Dial: 206.217.2226

AXIOS LAW GROUP
1525 Fourth Avenue, Suite 800
Seattle, WA 98101
Telephone: 206.217.2200
Customer No.: 61857

IX. CLAIMS APPENDIX A

1. (Previously Presented) A method of storing streamed presentation data within a container file, the method comprising:

receiving one or more data streams from each of one or more presentation sources within the presentation;

creating within the container file, a virtual file for each of the one or more presentation sources;

temporarily storing first data associated with a first data stream of a first presentation source in association with a first virtual file corresponding to the presentation source;

determining a container file size of the container file;

temporarily storing additional data from the first data stream in place of at least a portion of the first data if the container file size is within a predetermined range of an identified maximum buffer size; and

rendering at least one of said one or more data streams.

2. (Original) The method of claim 1, wherein the additional data from the first data stream is stored in place of at least a portion of the first data if the container file size is equal to or exceeds the identified maximum buffer size.

3. (Original) The method of claim 1, further comprising: temporarily storing second data associated with a second data stream of the first presentation source in association with the first virtual file; and temporarily storing additional data from the second data stream in place of at least a portion of the second data stored in association with the first virtual file if the container file size is within the predetermined range of the identified maximum buffer size.

4. (Original) The method of claim 3, further comprising: rendering one of the first and second data streams in real-time contemporaneous with the storing of at least one of the first and second data streams.

5. (Original) The method of claim 3, further comprising: temporarily storing data associated with a third data stream of a second presentation source in association with a second virtual file; and temporarily storing additional data from the third data stream in place of at least a portion of the data stored in association with the second virtual file if the container file size is within the predetermined range of the identified maximum buffer size.

6. (Original) The method of claim 1, wherein the maximum buffer size is proportional to an amount of time indicated via a user interface.

7. (Original) The method of claim 1, wherein the maximum buffer size is dynamically increased during the storing of data from the first data stream.

8. (Original) The method of claim 1, wherein the first data and additional data are stored in a native packet format prior to a decoding process.

9. (Original) The method of claim 1, wherein each virtual file comprises: at least a first data block; and a file descriptor block containing at least a seek index and a seek index granularity, wherein the seek index indicates a plurality of equally distributed data blocks within the corresponding virtual file and the granularity indicates a size for each of the data blocks.

10. (Original) The method of claim 9, wherein the additional data is stored in place of the first data beginning with the first data block and continuing with successive data blocks of the first virtual file.

11. (Original) The method of claim 9, wherein if the container file size is within the predetermined range of the identified maximum buffer size, the seek index granularity is increased so as to increase data block size without changing the number of seek index entries.

12. (Original) The method of claim 9, further comprising: receiving a user indication identifying a location corresponding to a time (T) within the presentation; identifying a seek position for each virtual file, wherein each seek position is determined by dividing time (T)

by the seek granularity for the corresponding virtual file; and contemporaneously rendering in real-time, data stored in each virtual file at the respective seek positions.

13. (Previously Presented) A machine readable medium having machine executable instructions, which when executed operate to implement a method comprising:

receiving one or more data streams from each of one or more presentation sources within a presentation;

creating within a container file, a virtual file for each of the one or more presentation sources;

temporarily storing first data associated with a first data stream of a first presentation source in association with a first virtual file corresponding to the presentation source;

determining a container file size of the container file;

temporarily storing additional data from the first data stream in place of at least a portion of the first data if the container file size is within a predetermined range of an identified maximum buffer size; and

rendering at least one of said one or more data streams.

14. (Original) The machine readable medium of claim 13, wherein the additional data from the first data stream is stored in place of at least a portion of the first data if the container file size is equal to or exceeds the identified maximum buffer size.

15. (Original) The machine readable medium of claim 13, further comprising instructions to temporarily store second data associated with a second data stream of the first presentation source in association with the first virtual file; and temporarily store additional data from the second data stream in place of at least a portion of the second data stored in association with the first virtual file if the container file size is within the predetermined range of the identified maximum buffer size.

16. (Original) The machine readable medium of claim 15, further comprising instructions to render one of the first and second data streams in real-time contemporaneous with the storing of at least one of the first and second data streams.

17. (Original) The machine readable medium of claim 15, further comprising instructions to: temporarily store data associated with a third data stream of a second presentation source in association with a second virtual file; and temporarily store additional data from the third data stream in place of at least a portion of the data stored in association with the second virtual file if the container file size is within the predetermined range of the identified maximum buffer size.

18. (Original) The machine readable medium of claim 13, wherein the maximum buffer size is proportional to an amount of time indicated via a user interface.

19. (Original) The machine readable medium of claim 13, wherein the maximum buffer size is dynamically increased during the storing of data from the first data stream.

20. (Original) The machine readable medium of claim 13, wherein the first data and additional data are stored in a native packet format prior to a decoding process.

21. (Original) The machine readable medium of claim 13, wherein each virtual file comprises: at least a first data block; and a file descriptor block containing at least a seek index and a seek index granularity, wherein the seek index indicates a plurality of equally distributed data blocks within the corresponding virtual file and the granularity indicates a size for each of the data blocks.

22. (Original) The machine readable medium of claim 21, wherein the additional data is stored in place of the first data beginning with the first data block and continuing with successive data blocks of the first virtual file.

23. (Original) The machine readable medium of claim 21, wherein if the container file size is within the predetermined range of the identified maximum buffer size, the seek index granularity is increased so as to increase data block size without changing the number of seek index entries.

24. (Original) The machine readable medium of claim 21, further comprising instructions to receive a user indication identifying a location corresponding to a time (T) within the presentation; identify a seek position for each virtual file, wherein each seek position is determined by dividing time (T) by the seek granularity for the corresponding virtual file; and contemporaneously render in real-time, data stored in each virtual file at the respective seek positions.

X. EVIDENCE APPENDIX

NONE.

XI. RELATED PROCEEDINGS APPENDIX

NONE.